第19卷第24期
2007年12月

系 统 仿 真 学 报©
Journal of System Simulation

Vol. 19 No. 24
Dec., 2007

阶梯型粒子群算法及在函数优化中的应用

陈得宝[1,2]，赵春霞[2]

（1.淮北煤炭师范学院物理系，安徽淮北 235000；2.南京理工大学计算机科学与技术学院，江苏 南京 210094）

摘　要：提出一种自适应动态群粒子群方法，根据粒子群的多样性，**采用梯形规律动态调整粒子群的规模**，既保证每个粒子都得到充分的进化，又保持了群体的多样性，使局部收敛的可能性大大减少。**此方法根据群体的多样性的大小，在减少群体规模时，采用较差淘汰法，淘汰一些较差的粒子，在增加粒子时，采用交叉法产生新个体，既保持粒子的继承性，又维持了粒子群的多样性。**对典型函数进行测试实验，结果与其它粒子群方法进行比较，验证了方法的有效性。

关键词：粒子群优化；函数优化；多样性函数；梯形粒子群(LPSO)

中图分类号：TP391.9　　**文献标识码：**A　　**文章编号：**1004-731X (2007) 24-5659-04

Ladder Particle Swarm Optimization and Application in Function Optimization

CHEN De-bao[1,2], *ZHAO Chun-xia*[2]

(1. Physical Department of Huaibei Coal Industry Teachers College, Huaibei 235000, China;
2. Computer Institute of Nanjing University of Science and Technology, Nanjing 210094, China)

Abatract: *An adaptive variable population method was proposed.* The population size was modulated based on *diversity function in type of ladder*, every particle was evaluated with adequate time and diversity of swarm was maintained, so the probability of local convergence was declined in large degree. Diversity of population plays a very important role in evolution, so in this method, *a number of particles with small fitness would be deleted when the size of current population should be reduced, and crossover operator is used to increase a number of new particles into swarm, the new swam inherits the old swarm and the diversity of population is maintained.* The effectiveness of the given algorithm is indicated by experiment of typical test function.

Key words: particle swarm optimization; function optimization; diversity function; ladder particle swarm optimization (LPSO)

引　言

粒子群算法(PSO)是模拟鸟群捕食行为而提出的一种基于群体特性的进化方法[1]，此方法有着粒子个数较少，计算简单，鲁棒性好，速度较快等优点，在许多优化问题中发挥了重要的作用[2,3]。作为一种群体智能优化方法，粒子群算法也具有局部收敛，进化后期由于群体的多样性差，收敛速度慢，精度较差等缺点。目前为止，出现了许多改进其性能的方法，大体可分为三大类，第一类是从改善粒子群的操作参数出发来改善粒子群的性能，其中代表性的方法是线性下降惯性权方法[4]；第二类是 PSO 与其它进化方法结合的研究，X.H.Shi 等将 GA 与 PSO 结合，实现了一种混合算法[5]；Natsuki 设计了一种高斯变异的粒子群算法[6]，还有如模拟退火粒子群算法[7]等；第三类是离散粒子群方法和研究粒子群结构的方法[8,9]。

在上述的方法中，粒子群均采用固定的群体规模，在进化过程中，粒子数量不变。而在进化计算方法中，群体规模的大小对群体收敛性能有着很重要的影响，过大的群体规模使算法的运算量增加，影响着算法的收敛速度，过小的群体规模有时使算法不收敛。为研究群体规模对进化计算性能的影响，出现了一系列动态群体进化计算方法，如微种群GA(μGA)[10]，此方法一般选择较小的种群规模，当群体处于收敛状态时，重新产生一些新的个体替换掉较差的个体，保留较好的个体；文献[11]提出一种锯齿形遗传算法，在群体规模减少到最小时，利用随机重新初始化方法，使群体规模回到初始值。这种随机补充个体的方法，前后代之间继承性受到一定的影响，而且由于每代必须有个体被淘汰，有些个体得不到充分进化。

本文提出一种阶梯型粒子群算法，在每个阶梯上对一定数量的群体进行一定代数的训练，在每个阶梯末对群体的多样性进行计算，根据多样性的大小动态地改变群体规模，采用删除较差个体的方法来减少粒子，用随机交叉方法增加粒子，既维持了群体信息的多样化，又减少粒子的冗余现象。通过对典型函数的测试实验，验证了方法的有效性。

1　阶梯型粒子群算法

1.1 粒子群算法

在传统的粒子群方法中，群体数量一般都是固定的，过大和过小数量的规模都会给粒子群的性能带来一定的影响。由于粒子群在飞翔的过程中，始终是跟随粒子到目前为止的最好位置和当前代的最好位置，如式(1),(2)描述[1]，且满足

收稿日期：2006-10-16　　**修回日期：**2007-03-06
基金项目：安徽省教育厅自然科学基金项目(2006KJ090B)
作者简介：陈得宝(1975-)，男，安徽安庆人，博士生，副教授，研究方向为人工智能、进化计算和机器人等；**赵春霞**(1964-)，女，北京人，博士，教授，博导，研究方向为研究方向为模式识别、图像处理、人工智能、机器人和进化计算等。

条件方程(3),(4)。粒子很容易收敛到某一个或几个局部最优点，因此维持粒子群的多样性对全局收敛有重要的意义。与其它进化方法一样，在进化的开始阶段种群数量较大，群体包含较多的信息，但在进化的后期，由于“齐次”性较好，继续保持大量的群体，只能是使计算量增加，对算法的收敛性能将不会有太大的提高。

$$V_i(k+1)=wV_i(k)+c_1rand(.)(X_{pbesti}(k)-X_i(k))+c_2rand(.)(X_{gbest}(k)-X_i(k)) \quad (1)$$

$$X_i(k+1)=X_i(k)+V_i(k+1) \quad (2)$$

$$\begin{aligned}&\text{if } V_i(k+1)>V_{\max} \text{ then } V_i(k+1)=V_{\max}\\&\text{else } V_i(k+1)<-V_{\max} \text{ then } V_i(k+1)=-V_{\max}\end{aligned} \quad (3)$$

$$\begin{aligned}&\text{if } X_i(k+1)>X_{\max} \text{ then } X_i(k+1)=X_{\max}\\&\text{else if } X_i(k+1)<X_{\min} \text{ then } X_i(k+1)=X_{\min}\end{aligned} \quad (4)$$

式(1)，(2)中，$X_i(K+1)$, $X_i(K)$分别为第 $K+1$ 时刻和第 K 时刻粒子 i 的位置，$V_i(K+1)$和 $V_i(K)$为 $K+1$ 时刻和 K 时刻粒子 i 的运动速度，w 为加权因子，其较大时有利于全局收敛，较小时有利于局部收敛，通常取 0.4 到 0.9 之间，c_1, c_2 为常数，影响着粒子更新过程中认知特性和社会特性的比例，X_{pbesti}和 X_{gbest} 分别为系统进化到当前代第 i 个个体的最好位置和粒子群中最优粒子的位置，$rand(.)$为[0，1]之间的随机数。式(3)中 $v_{\max}$ 为粒子运动的最大速度，式(4)中 $x_{\max}, x_{\min}$ 为粒子运动的最大位置和最小位置。

1.2 阶梯型粒子群算法设计

1.2.1 阶梯型结构设计

在设计阶梯型方法之前，首先给定最大进化代数，记为 $G_{\max}$。如图 1，将最大进化代数分成若干相等的阶段，记为 T, 不同阶梯上的群体数量由多样性函数决定，即在每个阶梯的结束时刻，计算此时群体的多样性，根据下文设计的增加和减少粒子数量原则来确定下一阶梯上需要的群体数量。

图 1 中 N_{min}, N_{av}, N_{max} 分别为给定的最小群体规模，平均规模和初始时的最大规模，通常最大进化代数 $G_{\max}=R*T$, 其中 $R\in Z$。

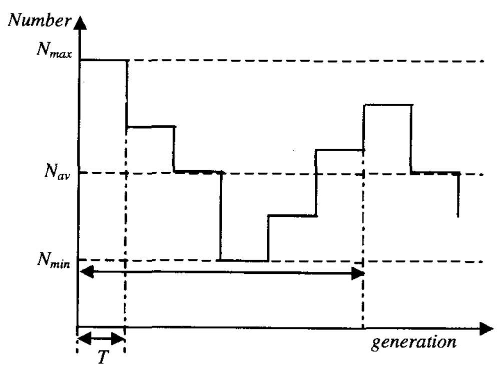

图 1　群体变化图

1.2.2 计算粒子的得分并排序

采用 q 竞争选择原理，即从这群体中随机抽取 q 个个体，对每个粒子，将其适应值分别与这 q 个个体比较，计算出 q 个个体中比已知个体差的个数 s_k, 并将其作为所选个体的得分，待所有粒子都比较完后，按得分高低顺序对粒子进行排序，在比较过程中，对每个比较都要重新采样，使得优良粒子的得分相对较高。

1.2.3 多样性函数设计

由于在粒子群算法中多样性的维持对算法的收敛影响较大，因此有必要设计一种衡量粒子群多样性的方法。

定义 1： 某一阶梯上，个体 g_i 与 g_k 之间的海明距离为：

$$H(g_i,g_k)=\sum_{j=1}^{L}\left|g_{ij}-g_{kj}\right| \quad (5)$$

其中 L 为个体基因长度。

定义 2： T 时间结束时，群体的平均海明距离为：

$$\overline{AH}(T)=\sum_{i=1}^{N(T)-1}\sum_{k=i+1}^{N(T)}H(g_i,g_k)/\sum_{i=1}^{N(T)-1}(N(T)-i) \quad (6)$$

其中 $N(T)$为群体中个体的数量。

因为个体之间的差异是衡量群体多样性是否优良的关键因素，设计多样性函数如下：

$$div(T)=\frac{1}{\pi}(a\tan(\overline{AH})+\frac{\pi}{2}) \quad (7)$$

由式(7)知，当粒子群的平均海明距离较大时，粒子间的差异较大，多样性函数值较大，反之则小，且多样性函数值被置为[0，1]。

1.2.4 增加和减少粒子数量的方法

通常，当群体的平均海明距离较大时，说明种群的多样性好，可以淘汰较多的个体，而当平均海明距离较小时，淘汰的个体数量相对较少。故首先要设计一个多样性阈值 th_{div}，本文采用比较两个连续阶梯上的多样性函数值的方法，若二者之差超过某一范围，取后者作为多样性函数的阈值，子程序流程如图 2。

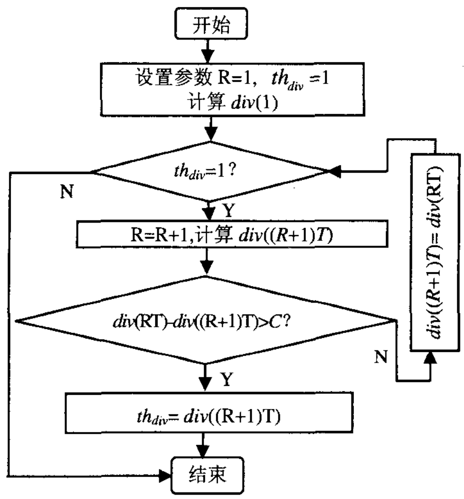

图 2　计算多样性阈值流程图

当多样性函数值大于 th_{div} 时，淘汰掉一部分粒子，设计每个阶梯保留的个体数量为：

$$N(T+1)=N(T)-N(T)*\eta*div(T) \quad (8)$$

其中 $N(T+1), N(T)$ 分别为 $T+1$，T 时间内的个体数量，

$\eta \in (0.1, 0.2)$ 为淘汰系数。将个体按适应度排序，按照式(6)，淘汰一定数量的较差的个体，当多样性函数值小于等于阈值 th_{div}，淘汰过程结束，如果此时的最优结果不满足要求，则需要补充个体，以增加种群的多样性。按与淘汰个体的方法类似，也以多样性来决定需要补充个体的数量，由于淘汰过程中希望淘汰较差的个体，而在增加种群多样性时，应该既保证种群多样性的增加，又希望新增加的个体对现有种群有一定的继承性。采用单点交叉方法[12]产生新个体，既在当前个体中选择一定数量的个体，按交叉方法，产生一定量的新个体补充到群体中。另外由于补充个体和淘汰个体的幅度不一样，淘汰的幅度不宜太大，否则会使种群瞬间变成齐次，增长的幅度可以大一些，能保证良好的多样性，因此增加个体数量采用下列方法：

$$N(T+1) = N(T) + N(T) * \lambda * div(T) \tag{9}$$

式(9)中 $\lambda \in [0.2, 1.0]$ 为增长算子。如果得到的个体数量大于初始化时的个体数量，则取此时的个体数量等于初始化时的个体数量，以保证其运算量的要求。

3 算法的实现过程描述

综合上述的分析，阶梯形粒子群算法的流程图如图 3。特别注意在图 3 中，一旦得到多样性阈值，在本次实验的其它代和阶梯的计算中将不再改变。

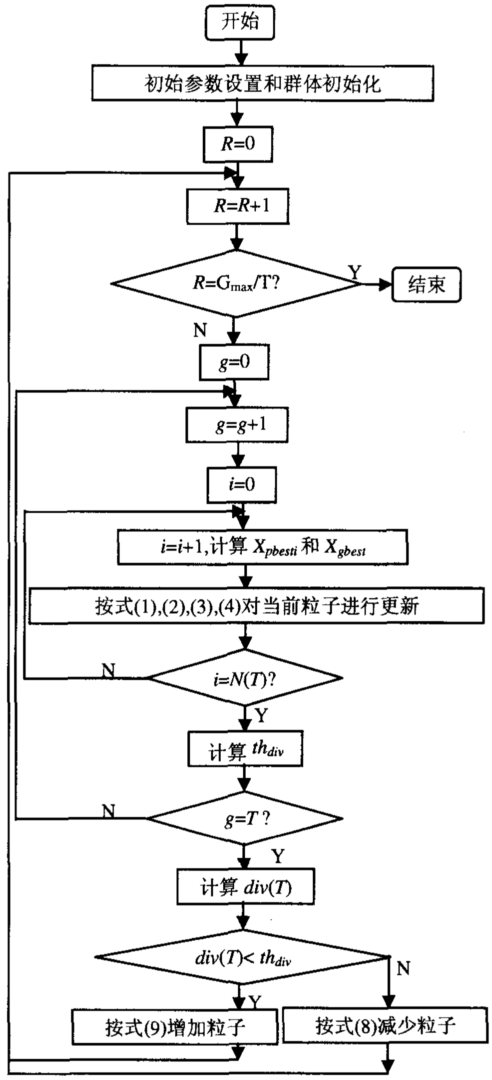

图 3 阶梯型粒子群算法流程图

4 实验

为验证此方法的有效性，对典型函数进行仿真实验，函数如下：

Rosenbrock function：

$$F_{Ros}(\vec{x}) = \sum_{i=1}^{n} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \quad -30 \le x_i \le 30$$

4.1 实验参数的设定

实验维数分别为 10，20，30，$c_1 = c_2 = 2$，R=40，T=50，本文方法的计算量略大于群体规模为其平均数量的固定群体规模粒子群算法，分析如下：

假设一个个体按照普通的粒子群飞翔方法的计算量为 c，排序和计算个体得分的运算量为 c_p，则在 R 个 T 时间内总的计算量为：

$$\begin{aligned} c_z &\approx (c + c_p) * (N(T) + N(2T) + \cdots + N(RT)) \\ &= (c + c_p) N_{av} \end{aligned} \tag{10}$$

由式(10)可看出，阶梯型粒子群算法的运算量略大于群体数量为其平均数量的普通粒子群算法的计算量，而实际中，影响计算速度的主要因素是乘除法运算，计算得分和排序占用计算机的时间很少。为比较两种方法的性能，实验中，本文方法初始粒子群规模为 100，固定规模方法的粒子群数量为 90。最大进化代数 2000，自变量的范围[-30,30]。w_{max}=0.9，w_{min}=0.4，采用文献[4]中的线性下降惯性权，即：

$$w = w_{max} - gen * \frac{w_{max} - w_{min}}{G_{max} / R} \tag{11}$$

4.2 实验结果与分析

分别用四种方法进行 20 次实验，所得到的最好适应度值和最差的适应度值如表 1 所示。20 次实验中，为节省篇幅，仅给出了粒子维数为 30 时的各函数实验的平均最优适应度如图 4，图中同时给出了采用线性下降惯性权粒子群方

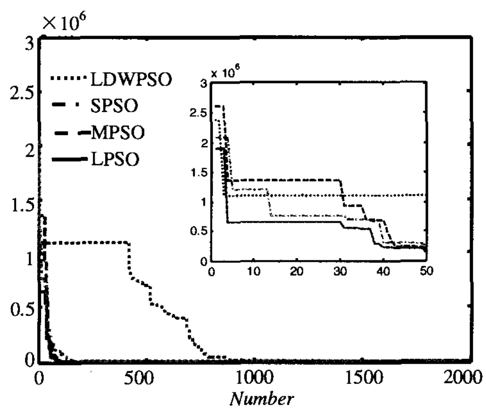

图 4 四种方法的平均适应度变化图

法[4](LDWPSO)，模拟退火粒子群方法[7](SPSO)，变异粒子群方法[6](MPSO)的结果。本文方法中，随机抽取某次实验群体规模变化如图 5。由表 1 可知，梯形粒子群方法能获得较好的适应值。由图还可看出本文方法在收敛速度和收敛精度上比其它方法好。

表 1 四种方法所得的最好和最差解

DIM	LPSO		LDWPSO		SPSO		MPSO	
	best	worst	best	worst	best	worst	best	worst
10	0.02	2.74	1.99	5.89	0.06	3.86	0.08	3.72
20	0.15	3.96	8.95	16.90	0.37	6.72	0.57	14.25
30	0.81	17.78	13.94	42.64	2.39	22.59	4.87	25.41

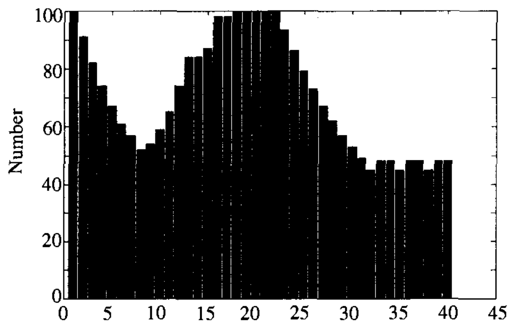

图 5 群体数量变化图

5 结论

本文提出一种体型群体规模的粒子群方法，粒子群根据设计的增减方法，根据多样性函数，自动更新自己的规模，在每一个阶梯上按照传统的方法进行更新，及保证每个粒子都得到充分的进化，又维持了粒子群的多样性，通过对典型函数的对比实验，验证了方法的有效性。

参考文献：

[1] J Kennedy, R C Eberhart. Particle Swarm optimization [C]// Proc. IEEE international Conference on Neural Network. USA: IEEE Press, 1995, 4: 1942-1948.

[2] J H Seo, C H Im, C G Heo, *et al.* Multimodal Function Optimization Based on Particle Swarm Optimization [J]. IEEE Trans. On Magnetics (S0018-9464), 2006, 42(4): 1095-1098.

[3] 高鹰，谢胜利，许若宁，等. 基于粒子群优化算法的稀疏信号盲分离[J]. 系统仿真学报，2006, 18(8): 2264-2266. (GAO Ying, XIE Sheng-li, XU Ruo-ning, LI Zhao-hui, *et al.* Blind Sparse Source Separation Based on Particle Swarm Optimization [J]. Journal of System Simulation, 2006, 18(8): 264-2266.)

[4] Y Shi, R C Eberhart. A modified Swarm Optimizer [C]// Proceedings of IEEE International Conference on Evolutionary Computation. NJ: IEEE Press, Piscataway, 1998: 69-73.

[5] X H Shi, Y C Liang, H P Lee, C Lu, L M Wang. An improved GA and a novel PSO-GA-based hybrid algorithm [J]. Information Processing Letters (S0020-0190), 2005, 93(5): 255-261.

[6] Natsuki Higasshi, Hitoshi Iba. Particle swarm optimization with Gaussian mutation [C]// Proceedings of the IEEE Swarm Intelligence Symp. Indianapolis: IEEE Inc., 2003:72-79

[7] 窦全胜，周春光，马铭. 粒子群优化的两种改进策略[J]. 计算机研究与发展，2005, 42(5): 897-904. (Dou Quansheng, Zhou Chunguang, Ma Ming. Two Improvement Strategies for Particle Swarm Optimization [J]. Journal of Computer Research and Development, 2005, 42(5): 897-904.)

[8] M Clerc, J Kennedy. The Particle swarm-Explosion, stability and convergence in a multi-dimensional complex space [J]. IEEE Trans. Evol.Comput. (S1089-778X), 2002, 6(1): 58-73.

[9] Y Shi, R C Eberhart. A modified Swarm Optimizer [C]// Proceedings of IEEE International Conference on Evolutionary Computation. NJ: IEEE Press, Piscataway, 1998: 69-73.

[10] F Fernóndez, M Tomassini, L Vanneschi. Saving Computational Effort in Genetic Programming by means of Plagues [C]// R Sarker, R Reynolds, H Abbass, *et al.* editors, CEC-2003, pages 2042-2049. The 2003 Congress on Evolutionary Computation, (CEC'03), USA: IEEE Press: Sarker, R Reynolds, H Abbass, *et al*, 2003: 2042-2049.

[11] Koumousis V K, Katsaras C P. A Saw-Tooth Genetic Algorithm Combining the Effects of Variable Population Size and Reinitialization to Enhance Performance [J]. IEEE Trans. On Evolutionary Computation (S1089-778X), 2006, 10(1): 19-28.

[12] D E Goldberg. Genetic Algorithms in Search, Optimization and MachineLearning [M]. Reading, MA: Addision-Wesley, 1989.